

Amendments to the Claims

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

- 1 1. (Currently Amended) A method executed in a computer system for processing
2 software source code with a variable looping statement having a loop index, an
3 initial expression, an exit expression and a body, wherein at least one of the
4 initial and exit expressions depends on a variable other than the loop index, to
5 enable loop unrolling, comprising:
6 determining from the initial expression and the exit expression, an a
7 constant upper bound and a;
8 determining from the initial expression and the exit expression, a constant
9 lower bound for a loop index within said variable looping statement;
10 determining a condition that must be satisfied, said condition reflecting any
11 conditions within ~~an~~ the initial expression and ~~an~~ the exit expression ~~of said~~
12 ~~variable looping statement;~~ and
13 forming replacing the variable looping statement in the software source
14 code with a constant looping statement, wherein said upper bound and said
15 lower bound define a range of values for a loop index within said constant
16 looping statement, wherein said constant looping statement includes a nested
17 conditional statement which tests said determined condition, wherein a body of
18 said constant looping statement comprises a the body of said variable looping
19 statement, and wherein said body of said constant looping statement is only
20 executed in the event that said determined condition is satisfied.
- 1 2. (Previously presented) The method of claim 1, wherein said determining said
2 condition comprises forming a logical "AND" of an initial condition within said
3 initial expression of said variable looping statement and an exit condition within
4 said exit expression of said variable looping statement.

- 1 3. (Original) The method of claim 1, further comprising determining whether said
2 variable looping statement includes an increasing loop index value.
- 1 4. (Original) The method of claim 3, further comprising:
2 in the event that said variable looping statement includes said increasing
3 loop index value, said determining of said lower bound comprises determining a
4 lower bound of said initial expression of said variable looping statement.
- 1 5. (Original) The method of claim 3, further comprising:
2 in the event that said variable looping statement includes said increasing
3 loop index value, said determining of said upper bound comprises determining an
4 upper bound of said exit expression of said variable looping statement.
- 1 6. (Original) The method of claim 1 further comprising determining whether said
2 variable looping statement includes a decreasing loop index value.
- 1 7. (Original) The method of claim 6, further comprising:
2 in the event that said variable looping statement includes said decreasing
3 loop index value, said determining of said lower bound comprises determining a
4 lower bound of said exit expression of said variable looping statement.
- 1 8. (Original) The method of claim 6, further comprising:
2 in the event that said variable looping statement includes said decreasing
3 loop index value, said determining of said upper bound comprises determining an
4 upper bound of said initial expression of said variable looping statement.
- 1 9. (Currently Amended) A system for processing software source code with a
2 variable looping statement having a loop index, an initial expression, an exit
3 expression and a body, wherein at least one of the initial and exit expressions
4 depends on a variable other the loop index to enable loop unrolling, said system

including a computer readable memory having one or more computer instructions stored thereon, said instructions comprising:

instructions operative to determine from the initial expression and the exit expression, an a constant upper bound and a;

determining from the initial expression and the exit expression a constant lower bound ~~for a loop index within said variable looping statement;~~

instructions operative to determine a condition that must be satisfied, said condition reflecting any conditions within ~~an~~ the initial expression and ~~an~~ the exit expression ~~of said variable looping statement;~~ and

instructions operative to ~~form~~ replace the variable looping statement in the software source code with a constant looping statement, wherein said upper bound and said lower bound define a range of values for a loop index within said constant looping statement, wherein said constant looping statement includes a nested conditional statement which tests said determined condition, wherein a body of said constant looping statement comprises a the body of said variable looping statement, and wherein said body of said constant looping statement is only executed in the event that said determined condition is satisfied.

10. (Previously presented) The system of claim 9, wherein said instructions operative to determine said condition comprise instructions operative to form a logical "AND" of an initial within said initial expression condition of said variable looping statement and an exit condition within said exit expression of said variable looping statement.

11. (Original) The system of claim 9, further comprising instructions operative to determine whether said variable looping statement includes an increasing loop index value.

12. (Original) The system of claim 11, further comprising: instructions operative, in the event that said variable looping statement includes said increasing loop index

value, to determine said lower bound by determining a lower bound of said initial expression of said variable looping statement.

13. (Original) The system of claim 11, further comprising: instructions operative, in the event that said variable looping statement includes said increasing loop index value, to determine said upper bound by determining an upper bound of said exit expression of said variable looping statement.

14. (Original) The system of claim 9 further comprising instructions operative to determine whether said variable looping statement includes a decreasing loop index value.

15. (Original) The system of claim 14, further comprising:
instructions operative, in the event that said variable looping statement includes said decreasing loop index value, to determine said lower bound by determining a lower bound of said exit expression of said variable looping statement.

16. (Original) The system of claim 14, further comprising: instructions operative, in the event that said variable looping statement includes said decreasing loop index value, to determine said upper bound by determining an upper bound of said initial expression of said variable looping statement.

17. (Currently Amended) A computer program product including a computer readable medium, said computer readable medium having a computer program stored thereon, said computer program for processing software source code with a variable looping statement having a loop index, an initial expression, an exit expression and a body, wherein at least one of the initial and exit expressions depends on a variable other than the loop index to enable loop unrolling, said computer program comprising:

8 program code for determining from the initial expression and the exit
 9 expression, an a constant upper bound and a;
 10 program code for determining from the initial expression and the exit
 11 expression, a constant lower bound for a loop index within said variable looping
 12 statement;
 13 program code for determining a condition that must be satisfied, said
 14 condition reflecting any conditions within ~~an~~ the initial expression and ~~an~~ the exit
 15 expression ~~of said variable looping statement;~~ and
 16 program code for ~~forming~~ replacing the variable looping statement in the
 17 software source code with a constant looping statement, wherein said upper
 18 bound and said lower bound define a range of values for a loop index within said
 19 constant looping statement, wherein said constant looping statement includes a
 20 nested conditional statement which tests said determined condition, wherein a
 21 body of said constant looping statement comprises a body of said variable
 22 looping statement, and wherein said body of said constant looping statement is
 23 only executed in the event that said determined condition is satisfied.

1 18. (Currently Amended) A computer data signal embodied in a carrier wave, said
 2 computer data signal including a computer program, said computer program for
 3 processing software source code with a variable looping statement having a loop
 4 index, an initial expression, an exit expression and a body, wherein at least one
 5 of the initial and exit expressions depends on a variable other than the loop index
 6 to enable loop unrolling, said computer program comprising:

7 program code for determining from the initial expression and the exit
 8 expression, an a constant upper bound and a;
 9 program code for determining from the initial expression and the exit
 10 expression a constant lower bound for a loop index within said variable looping
 11 statement;
 12 program code for determining a condition that must be satisfied, said
 13 condition reflecting any conditions within ~~an~~ the initial expression and ~~an~~ the exit
 14 expression ~~of said variable looping statement;~~ and

15 program code for ~~forming~~ replacing the variable looping statement in the
16 software source code a constant looping statement, wherein said upper bound
17 and said lower bound define a range of values for a loop index within said
18 constant looping statement, wherein said constant looping statement includes a
19 nested conditional statement which tests said determined condition, wherein a
20 body of said constant looping statement comprises a the body of said variable
21 looping statement, and wherein said body of said constant looping statement is
22 only executed in the event that said determined condition is satisfied.

- 1 19. (Currently Amended) A system for processing software source code with a
2 variable looping statement having a loop index, an initial expression, an exit
3 expression and a body, wherein at least one of the initial and exit expressions
4 depends on a variable other than the loop index, to enable loop unrolling,
5 comprising:
6 means for determining from the initial expression and the exit expression,
7 an a constant upper bound and a;
8 means for determining from the initial expression and the exit expression a
9 constant lower bound for a loop index within said variable looping statement;
10 means for determining a condition that must be satisfied, said condition
11 reflecting any conditions within ~~an~~ the initial expression and ~~an~~ the exit
12 expression ~~of said variable looping statement;~~ and
13 means for ~~forming~~ replacing the variable looping statement in the software
14 source code with a constant looping statement, wherein said upper bound and
15 said lower bound define a range of values for a loop index within said constant
16 looping statement, wherein said constant looping statement includes a nested
17 conditional statement which tests said determined condition, wherein a body of
18 said constant looping statement comprises a the body of said variable looping
19 statement, and wherein said body of said constant looping statement is only
20 executed in the event that said determined condition is satisfied.